Year: 2016

Multi-body motion estimation from monocular vehicle-mounted cameras

Sabzevari, Reza ; Scaramuzza, Davide

Abstract: This paper addresses the problem of simultaneous estimation of the vehicle ego-motion and motions of multiple moving objects in the scene—called eoru-motions—through a monocular vehicle-mounted camera. Localization of multiple moving objects and estimation of their motions is crucial for autonomous vehicles. Conventional localization and mapping techniques (e.g. Visual Odometry and SLAM) can only estimate the ego-motion of the vehicle. The capability of robot localization pipeline to deal with multiple motions has not been widely investigated in the literature. We present a theoretical framework for robust estimation of multiple relative motions in addition to the camera ego-motion. First, the framework for general unconstrained motion is introduced and then, it is adapted to exploit the vehicle kinematic constraints to increase efficiency. The method is based on projective factorization of the multiple-trajectory matrix. First, the ego-motion is segmented and, then, several hypotheses are generated for the eoru-motions. All the hypotheses are evaluated and the one with the smallest reprojection error is selected. The proposed framework does not need any a priori knowledge of the number of motions and is robust to noisy image measurements. The method with constrained motion model is evaluated on a popular street-level image dataset collected in urban environments (KITTI dataset) including several relative ego-motion and eoru-motion scenarios. A benchmark dataset (Hopkins 155) is used to evaluate this method with general motion model. The results are compared with those of the state-of-the-art methods considering a similar problem, referred to as the Multi-Body Structure from Motion in the computer vision community.

DOI: https://doi.org/10.1109/TRO.2016.2552548

Originally published at:
Sabzevari, Reza; Scaramuzza, Davide (2016). Multi-body motion estimation from monocular vehicle-mounted cameras. IEEE Transactions on Robotics, 32(3):638-651.
DOI: https://doi.org/10.1109/TRO.2016.2552548

Multi-body Motion Estimation from Monocular Vehicle-Mounted Cameras

Reza Sabzevari *Member, IEEE,* and Davide Scaramuzza *Member, IEEE,*

***Abstract*—This paper addresses the problem of simultaneous estimation of the vehicle *ego-motion* and motions of multiple moving objects in the scene—called *eoru-motions*—through a monocular vehicle-mounted camera. Localization of multiple moving objects and estimation of their motions is crucial for autonomous vehicles. Conventional localization and mapping techniques (e.g. Visual Odometry and SLAM) can only estimate the ego-motion of the vehicle. The capability of robot localization pipeline to deal with multiple motions has not been widely investigated in the literature. We present a theoretical framework for robust estimation of multiple relative motions in addition to the camera ego-motion. First, the framework for general unconstrained motion is introduced and then, it is adapted to exploit the vehicle kinematic constraints to increase efficiency. The method is based on projective factorization of the *multiple-trajectory matrix*. First, the *ego-motion* is segmented and, then, several hypotheses are generated for the *eoru-motions*. All the hypotheses are evaluated and the one with the smallest reprojection error is selected. The proposed framework does not need any a priori knowledge of the number of motions and is robust to noisy image measurements. The method with constrained motion model is evaluated on a popular street-level image dataset collected in urban environments (*KITTI* dataset) including several relative *ego-motion* and *eoru-motion* scenarios. A benchmark dataset (*Hopkins 155*) is used to evaluate this method with general motion model. The results are compared with those of the state-of-the-art methods considering a similar problem, referred to as the *Multi-Body Structure from Motion* in the computer vision community.**

MULTIMEDIA MATERIAL

This paper is accompanied by a video available on the author webpage.

I. INTRODUCTION

Visual odometry is the process of estimating the motion of a vehicle using only observations from its onboard cameras [1]. The vehicle motion is estimated under the assumption that the world is predominantly static; thus, moving objects are normally treated as outliers. The problem of estimating the motion of other moving objects in the scene is known as Multi-Body Structure from Motion (MBSfM).

In this paper, we propose a theoretical framework for the problem of estimating the vehicle *ego-motion* along with the other bodies' motions—so-called *eoru-motion*—observed by a car-mounted camera in *urban environments*, cf. Fig. 1. We first introduce the framework for general unconstrained motion and then adapt it to vehicle-mounted cameras, by exploiting the vehicle kinematic constraints, in order to increase the algorithm efficiency. The proposed method is inspired by series of works on MBSfM and can be seen as a valid complement to standard Visual Odometry (VO) and visual Simultaneous Localization And Mapping (SLAM) pipelines. Possible applications are driver-assistance systems (e.g., to estimate the motions of other on-road objects) and multi-robot collaboration scenarios [2], where a group of robots needs to work together to accomplish a given task.

Fig. 1: Estimated motions: ego-motion in *red* and eoru-motion in *yellow* and *blue*.

Our work targets monocular vision. Cameras are very cost effective, in terms of price, data transmission, and power consumption. The advantage of monocular vision over stereo vision is that the former scales well with both the environment and robot size.

A. Related Work

The problem of estimating multiple motions and structures from 2D correspondences is known as *Multi-body Structure from Motion* or *Motion Segmentation and Estimation*. The problem addressed differs from *Multi-Target Tracking*. The former deals with estimating the motions in 3D space and recovering the 3D structures; the latter deals with tracking the objects in the image plane; thus, estimated motions are 2D vectors on the image plane.

The works on MBSfM can be categorized into two major groups, depending on whether they use a perspective camera model or not (i.e., affine and orthografic). Solving this problem for perspective cameras is more challenging than affine or orthographic projection as the projective depth scales are

The authors are with the Robotics and Perception Group, University of Zurich, Switzerland. This research was supported by the Hasler Foundation (project number 13027), the UZH Forschungskredit, and the Swiss National Science Foundation through the National Center of Competence in Research Robotics (NCCR).

also unknown. Murakami et al. [3] studied circumstances where a projective factorization is feasible without estimating projective depth values, and showed that is possible only under strict assumptions.

There are several works in the literature, which provide accurate motion segmentation and estimation under affine camera model. The seminal work of Costeira and Kanade [4] formulated the multi-body SfM problem as a factorization problem. Zappella et al. [5] used the same formulation for orthographic camera model in an optimization framework. Their method can handle missing entries in the trajectory matrix caused by loss of feature tracks in a few frames. Yan and Pollefeys [6] can fairly handle outliers. However, since their method estimates the subspaces locally, it is unable to handle cases where two or more parts of the scene have the same motion but are not spatially correlated.

On the contrary, MBSfM has not been well-studied for perspective images. Vidal et al. [7] proposed an algebraic approach to estimate multiple structures and motions from two perspective views. This work was then extended to three views in [8]. However, since both methods are based on geometric approaches, they are not robust to noise and, thus, cannot be used for real-world applications. Recently, Ji et al. [9] proposed a method (based on the notions of subspace clustering) to perform motion segmentation without any knowledge about point correspondences across images. They formulated the problem in terms of Partial Permutation Matrices to match feature descriptors while satisfying subspace constraints for point trajectories. Schindler et al. [10] proposed a method for n-view multi-body SfM based on model selection. Their method uses 2-view geometry and, by linking motion segments between multiple pairs of frames, propagates the initial segmentation to n views. Differently, Li et al. [11] proposed a factorization approach to identify multiple rigid motions in perspective images. Their method is based on an initial estimation of projective depth scales and consequently is not robust to noise. Details of the perspective factorization approach to multi-body SfM are discussed in Section II.

Visual odometry is a well-defined problem, which has been largely studied in the literature. An exhaustive survey on VO is presented in [1], [12]. Among the works on VO, those exploiting the vehicle kinematics are relevant for this paper. Scaramuzza et al. [13], [14] leveraged the Ackerman-steering principle [15] to approximate the vehicle motion as locally-planar and circular and showed that this allows parametrizing the car motion in terms of a single feature correspondence. This led to very efficient algorithms for structure from motion, such as 1-point RANSAC or histogram voting. Since in real-world scenarios cars can violate the locally-planar and circular motion assumption, in [16], [17] the same authors relaxed this assumption and solved the relative structure from motion problem as a maximum-likelihood estimation problem using a locally planar and circular motion prior.

The opportunities that multi-body SfM provides to navigation algorithms have been rarely investigated in the literature. On the other hand, most of the experiments for MBSfM in the literature are based on synthetic datasets. One of the few works in this context was done by Vidal in [18], who applied subspace clustering techniques to motion segmentation in perspective images. However, the motion segmentation was applied on optical flow information of an outdoor sequence to segment the motions but not for estimating the motions. Recently, Kundu et al. [19] proposed an incremental framework for simultaneous reconstruction and segmentation for smoothly moving cameras. They used individual motion segmentation and reconstruction modules supported by a tracking module. In their work, the motion segmentation is done through a combination of epipolar and flow-vector-bound constrains in a probabilistic framework. The motion segmentation module provides priors for the reconstruction module and a particle-filter–based tracking is used for individual motions to estimate the 3D position and velocity of a moving target.

Vogel et al. [20] proposed a method to estimate dense scene flow from multiple pairs of stereo images (i.e. four temporal frames). The depth from disparity was used to extend the 2D optical flow to the 3D scene flow. The 3D scene flow can be exploited to segment multiple motions, and the depth from disparity can be used to initialize the 3D structures, given motion-segmentation. However, their work did not aim at segmenting the motions of different moving objects and estimate their 3D structures.

In a similar spirit, Rabe et al. [21] track interest points and fuse them with depth from stereo with a Kalman filter. They focused on estimating the 3D motion field in real-time, but not on segmenting motions or generating 3D structures of moving objects. Their work was extended in [22] using a 2.5D representation of the scene. They group pixels of the same depth to fixed width vertical stripes (called *Stixels*) in the image, as a mid-level representation of the world—in contrast to pixel-level and object-level representations. Each Stixel is individually tracked as in [21], but grouping the Stixels by segmenting their 3D motion is not considered. Badino and Kanade [23] also use a Kalman filter to fuse spatial and temporal information from a head-mounted stereo camera to simultaneously estimate the 3D position and the velocity of interest points in the 3D space. Similar to other real-time stereo approaches, they estimate the ego motion but their method is not aimed to group points moving together independently of the camera motion by segmenting the 3D motion field.

B. Contributions

This paper extends our previous work [24], where we first introduced our theoretical framework for simultaneous ego and eoru motion estimation for general 6-DoF motion of the camera. In this paper, we will first summarize this general framework and then adapt it to vehicle-mounted cameras. More specifically, instead of estimating full motion models, here we estimate minimal motion models by enforcing the constraints imposed by vehicle kinematics (i.e., nonholonomic constraints). Enforcing motion constraints decreases the number of parameters to estimate, thus making the algorithm to converge substantially faster.

Our method is based on the factorization of the multiple-trajectory matrix. However, unlike other MBSfM methods, which require an initial segmentation of motions, our method generates and evaluates several hypotheses for motion-segments. This makes the method more robust to noise and independent of any a priori assumption on the number of motions.

C. Paper Outline

In Section II, the theoretical background of single-body and multi-body SfM from perspective views is described. This part mainly focuses on solving such problems for rigid motions through factorization of the multiple-trajectory matrix. In Section III, the proposed framework and its theoretical concepts are presented. Then, the vehicle's kinematic constraints are introduced and integrated in the framework. In Section IV, results of the proposed approach on a street-level dataset [25] are presented, showing the performance of the proposed method using motion constraints imposed by urban environments. A benchmark dataset [26] is also used to evaluate the performance of proposed framework with generic unconstrained motion, and to compare it with previous works. Moreover, it is shown how the use of motion constraints affects the computational complexity of the algorithm compared with the case of general unconstrained motion.

II. Multi-Body Structure and Motion through Factorization

Structure from motion can be considered as the simultaneous solution of two dual problems: *i)* recovering an unknown structure from known camera positions, *ii)* determining the viewer's positions or camera motion from a set of known 2D points. In general, 3D structure and camera motion can be estimated by applying epipolar geometry between every pair of images or using multi-view geometric constraints. The inter-image relations are linked by the fact that a unique shape is projected onto the images captured from different views. Since the image correspondences are usually sparse 2D image points, the estimated 3D structure is also a sparse 3D point cloud.

Consider a set of $p \in \mathbb{N}$ 2D point correspondences in $f \in \mathbb{N}$ views accumulated in a matrix $\mathtt{W} \subset \Omega^f$, where $\Omega \subset \mathbb{R}^2$ is the image domain. Given the matrix $\mathtt{W}$, the SfM problem is solved simultaneously for the position of points in 3D space, denoted as $\mathtt{S} \subset \mathbb{R}^3$, and the relative poses of the cameras representing the motion, denoted as $\mathtt{M} \in SE(3) : (\mathbb{R}^3 \mapsto \Omega)^f$. A set of popular approaches (e.g. [4], [27], [28]) estimate $\mathtt{M}$ and $\mathtt{S}$ matrices via factorization methods using solely the collection of such 2D image point correspondences.

A. Rigid Structure and Motion: Perspective Camera Model

Estimation of structures and motions of rigid moving objects can be formulated in the mathematical context of bilinear matrix factorization. Therefore, the 2D image trajectories used by SfM can be described by bilinear matrix models [27]. In more detail, by defining the image coordinates of a point $i \in \mathbb{N}$ in frame $g \in \mathbb{N}$, for the case of the perspective camera model, we have:

$$\mathbf{w}_{gi} = \lambda_{gi}\ [x_{gi}\ \ y_{gi}\ \ 1]^\top = [u_{gi}\ \ v_{gi}\ \ \lambda_{gi}]^\top , \tag{1}$$

where vector $\mathbf{w}_{gi} \in \mathbb{R}^3$ denotes the homogeneous coordinates of the i^{th} point in the g^{th} image frame that is scaled by the corresponding projective depth value $\lambda_{gi} \in \mathbb{R}^+$. Thus, the measurement matrix $\mathtt{W}$ that gathers the corresponding 2D measurements in all views can be expressed as:

$$\mathtt{W} = \begin{bmatrix} \mathbf{w}_{11} & \dots & \mathbf{w}_{1p} \\ \vdots & \ddots & \vdots \\ \mathbf{w}_{f1} & \dots & \mathbf{w}_{fp} \end{bmatrix} , \tag{2}$$

where f is the number of frames ($g = 1 \dots f$) and p is the number of points ($i = 1 \dots p$). In case of a rigid object, the camera motion matrices $\bar{\mathtt{M}}_g$ and the 3D points $\mathbf{s}_i$ can be expressed as:

$$\bar{\mathtt{M}}_g = \left[\begin{array}{ccc|c} R_{g1} & R_{g3} & R_{g5} & t_{g1} \\ R_{g2} & R_{g4} & R_{g6} & t_{g2} \\ 0 & 0 & 0 & 1 \end{array}\right] \quad \text{and} \quad \mathbf{s}_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} , \tag{3}$$

where $\bar{\mathtt{M}}_g \in \mathbb{R}^{3\times 4}$ is the projection matrix for the g^{th} frame containing rotation and translation components and $\mathbf{s}_i$ is a 4-vector containing the homogeneous coordinates of the i^{th} point in 3D space. So, a 2D point i in a frame g is given by $\mathbf{w}_{gi} = \bar{\mathtt{M}}_g\, \mathbf{s}_i$.

We can collect all image measurements and their respective bilinear components $\bar{\mathtt{M}}_g$ and $\mathbf{s}_i$ in a global matrix form. Thus, the factorization model of image trajectories can be formulated as:

$$\mathtt{W}_{3f\times p} = \mathtt{M}_{3f\times 4}\ \ \mathtt{S}_{4\times p} , \tag{4}$$

where the bilinear components $\mathtt{M}$ and $\mathtt{S}$ are defined as:

$$\mathtt{M} = \begin{bmatrix} \bar{\mathtt{M}}_1 \\ \vdots \\ \bar{\mathtt{M}}_f \end{bmatrix} \quad \text{and} \quad \mathtt{S} = \begin{bmatrix} \mathbf{s}_1 & \cdots & \mathbf{s}_p \end{bmatrix} . \tag{5}$$

In general, the rank of $\mathtt{W}$ is constrained to $\operatorname{rank}\{\mathtt{W}\} \leq r$, where $r \ll \min\{3\times f,\ p\}$. In practice, the image measurements cannot be noise-free, which increases the rank of matrix $\mathtt{W}$. Thus, the rank-4 constraint should be enforced in the factorization.

Factorization of Eq. (4) with the rank-4 constraint is possible if the depth scales λ_{gi} are known. Using epipolar geometry, Sturm and Triggs [28] proposed a method to estimate λ_{gi} up to an arbitrary scale factor. This can be achieved by estimating the fundamental matrices $\mathtt{F}_{gg'}$ and, consequently, the epipoles $\mathbf{e}_{gg'}$ that relate every pair of consecutive frames g and g'. These two elements ($\mathtt{F}_{gg'}$ and $\mathbf{e}_{gg'}$) can be estimated in a *least-squares* manner using the 8-point algorithm [29]. Thus, the relation between depth scales λ_{gi} and $\lambda_{g'i}$ in two consecutive frames is:

$$\lambda_{gi} = \frac{\left(\mathbf{e}_{gg'} \times \mathbf{w}_{gi}\right)^\top \left(\mathtt{F}_{gg'}\mathbf{w}_{g'i}\right)}{\left\|\mathbf{e}_{gg'} \times \mathbf{w}_{gi}\right\|^2}\ \lambda_{g'i} . \tag{6}$$

Writing Eq. (6) for every pair of corresponding image points and every pair of consecutive image frames, the depth values can be recovered recursively up to an arbitrary initial value of λ_{1i}. In practice, the image measurements are noisy, and relying only on geometric estimations will not provide enough robustness. The robustness can be increased by iteratively alternating between two steps: *i)* rank-4 estimation of structure $\mathtt{S}$ and motion $\mathtt{M}$ matrices, given an initial estimate for depth values λ_{gi}, *ii)* estimating the depth values that improve the previous estimations of structure and motion [30]. In more detail, if the depth values are initialized as $\lambda_{gi} = 1$, then the best rank-4 estimation of $\mathtt{W}$ is:

$$\begin{aligned} \mathtt{W}_{3f\times p} &\approx \tilde{\mathtt{M}}_{3f\times 4}\,\tilde{\mathtt{S}}_{4\times p}, \\ \tilde{\mathtt{W}} &= \tilde{\mathtt{M}}\,\tilde{\mathtt{S}}, \end{aligned} \tag{7}$$

where $\tilde{\mathtt{S}}$ and $\tilde{\mathtt{M}}$ are the best rank-4 estimations for structure and motion, respectively, and $\tilde{\mathtt{W}}$ is an approximation of $\mathtt{W}$ given by $\tilde{\mathtt{S}}$ and $\tilde{\mathtt{M}}$. Once the estimations for motion and structure are obtained, the depth values are estimated as:

$$\lambda_{gi} = \left\| \mathbf{w}_{gi} - \tilde{\mathbf{w}}_{gi} \right\|, \tag{8}$$

where $\tilde{\mathbf{w}}_{gi}$ is an approximation of $\mathbf{w}_{gi}$ given by Eq. (7). Oliensis and Hartley [30] proved the convergence of such an iterative scheme.

B. From Single Motion to Multiple Motions

If the 2D image correspondences belong to motions of multiple objects, the image measurement matrix $\mathtt{W}$ that envelopes all image correspondences belonging to several motions can be written as:

$$\mathtt{W} = [\mathtt{W}_1|\mathtt{W}_2|\dots|\mathtt{W}_n], \tag{9}$$

where n is the number of motions and $\mathtt{W}_j$, $j = 1 \dots n$, is the matrix containing 2D point correspondences belonging to the j^{th} motion. Basically, matrix $\mathtt{W}$ is the horizontal concatenation of $\mathtt{W}_j$ matrices, each containing p_j points that comply with motion j, where $p = \sum_{j=1}^{n} p_j$ is the total number of points for all motions. So, the motion matrix $\mathtt{M}$ and structure matrix $\mathtt{S}$ can be written as:

$$\mathtt{M} = [\mathtt{M}_1|\mathtt{M}_2|\dots|\mathtt{M}_n] \quad \text{and} \quad \mathtt{S} = \begin{bmatrix} \mathtt{S}_1 & 0 & \dots & 0 \\ 0 & \mathtt{S}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathtt{S}_n \end{bmatrix}. \tag{10}$$

In this case, the generic SfM equation, $\mathtt{W} = \mathtt{M}\,\mathtt{S}$, is:

$$[\mathtt{W}_1|\dots|\mathtt{W}_n] = [\mathtt{M}_1|\dots|\mathtt{M}_n] \cdot \begin{bmatrix} \mathtt{S}_1 & & \\ & \ddots & \\ & & \mathtt{S}_n \end{bmatrix}. \tag{11}$$

For a perspective camera, matrix $\mathtt{W}$ belongs to $\mathbb{R}^{3f\times p}$ and matrix $\mathtt{W}_j \in \mathbb{R}^{3f\times p_j}$, both holding homogenous image coordinates scaled by depth values λ_{gi}. Consequently, matrix $\mathtt{M}$ is a $3f \times 4n$ matrix which contains individual motion matrices $\mathtt{M}_j \in \mathbb{R}^{3f\times 4}$. To recover multiple structures and motions, the sparse structure of $\mathtt{S}$ is employed and, using Eq. (11), the image measurement matrix $\mathtt{W}$ is factorized such that the noise in zero areas of matrix $\mathtt{S}$ is minimized. This can be achieved by iteratively alternating between estimating two components: *i)* the 3D structures by maximizing the sparsity of matrix $\mathtt{S}$, *ii)* the motion matrices by minimizing reprojection error and discarding the points from matrices $\mathtt{W}$ and $\mathtt{S}$ that cause a large reprojection error. Li et al. [11] proposed an approach for projective factorization of multiple rigid motions based on depth estimation method of Strum and Triggs [28]. In their method, an initial motion segmentation as well as an initial depth estimation are required. An iterative refinement stage alternates between estimating the depth values and motion segments. Once the motion segments and depth values are converged, motion and structure for each motion-segment are estimated via factorization.

III. Proposed Approach

In this section, the proposed approach for estimating relative motion and structure of independently moving objects is discussed. Given f perspective views of p points belonging to rigid objects moving under n classes of motions, the goal is to segment these points based on their motions, estimate motions and recover the position of the points in 3D coordinates.

In more detail, consider set $\mathscr{P} = \{P_1, \dots P_p\}$ containing indices for p point trajectories, such that:

$$\mathscr{P} = \bigcup_{j=1}^{n} \mathscr{P}_j, \tag{12}$$

where $\mathscr{P}_j$ is the set of point trajectories that obey motion j and ideally $\mathscr{P}_{j'} \bigcap \mathscr{P}_j = \emptyset$, where $j' \neq j$. Thus, set $\mathscr{P}_j$ will include p_j columns of matrix $\mathtt{W}$ (see Eq. (2)) such that:

$$\begin{aligned} \mathscr{P}_j &= \left\{ \overline{\mathbf{w}}_j^{(1)}, \dots, \overline{\mathbf{w}}_j^{(p_j)} \right\}, \\ \mathtt{W}_j &= \left[\overline{\mathbf{w}}_j^{(1)}, \dots, \overline{\mathbf{w}}_j^{(p_j)} \right], \end{aligned} \tag{13}$$

where matrix $\mathtt{W}_j$ contains all columns (i.e. $\overline{\mathbf{w}}_j^{(.)}$) of matrix $\mathtt{W}$ that have similar motions among the f frames.

Among the subsets of $\mathscr{P}$, there is always a subset of points that belongs to the camera motion. In other words, this subset of points represents the static parts of the scene, which is usually the dominant perceived motion. Since the camera is attached to wheeled vehicle with Ackermann steering, the motion is instantaneously planar and circular [31], and, therefore, can be parametrized by only two degrees of freedom. This allows the image point correspondences satisfying this motion to be segmented with the 1-point algorithm [14], which results in a very efficient segmentation of the camera movement. Let us call the segmented camera motion $\mathscr{P}_1$. Thus, there should be $n-1$ other motions to segment and estimate. Assuming that all moving objects in the scene have rigid planar motions—but not necessarily locally circular—the minimal solution for modeling such kind of motions is the 2-point algorithm [32]. Thus, all the motions in the scene can be categorized into two different types of motions: a *camera motion* that is modeled as a 1-DOF motion and a set of

Algorithm 1 Outline of Simultaneous Motion Segmentation and Reconstruction

Input: *2D image correspondences*
Output: *Motions and structures of independent rigid bodies*

1: Segment the *camera motion* using 1-point algorithm (*see III-A*)
2: Generate enough hypotheses for objects motions using 2-point algorithm (*see Alg. (2)*)
3: Evaluate each motion-segment hypothesis by computing reprojection error (*see Alg. (3)*)
4: **return** The structures and motions for the hypothesis with the smallest reprojection error

objects motions modeled as 2-DOF motions. Finding subsets of $\mathscr{P}$, holding Eq. (12), results in a motion segmentation hypothesis. Given ψ hypotheses for motion segments, they are evaluated by calculating the reprojection error with respect to all estimated motions and structures. In the evaluation phase, matrices $\mathtt{W}$, $\mathtt{S}$, and $\mathtt{M}$ as in Eq. (11) are shaped for every motion hypotheses. After initializing these matrices, the reprojection error is calculated and minimized by iteratively detecting the outliers for each motion-segment and verifying them with other motion-segments. The motion-segment hypothesis with the smallest reprojection error will be reported as the best one to describe the trajectory matrix $\mathtt{W}$. The outline of our algorithm is presented in Alg. (1).

A. Modeling Camera Ego-motion

To segment the 2D point correspondences belonging to camera motion with respect to static parts of the scene, it is assumed that the vehicle motion is locally (between two consecutive frames) planar and circular. In fact, considering the dynamics of the vehicle's contacts with the ground (during acceleration, break, slip, sharp turns, etc.) as well as dynamics of the suspension system, the vehicle motion is neither planar nor circular. On the other hand, current cameras with high frame rate can compensate the violation of vehicle's dynamics from locally planar and circular motion in most cases. This makes the assumption of locally planar and circular motion valid for the entire path—even for long trajectories—if images are captured at a high frequency [16]. As shown in [33], the required frame rate is ≥ 10 Hz, for a car driving at 50 km/h. This means, the baselines of images should be ≤ 1 m, in order to satisfy the locally planar and circular assumption. Thus, for any wheeled vehicle—on a short trajectory that satisfies the required baseline—there exists an instantaneous center of rotation C that describes the planar vehicle motion via a rotation angle θ (see Fig. 2), as discussed in [14]. The existence of such instantaneous center of rotation results from the Ackermann steering geometry [31]. This means that the motion model of the vehicle-mounted camera has only one degree of freedom. So, a single point correspondence is enough to estimate the motion, as demonstrated in [14].

Based on the Ackermann steering geometry, the vehicle motion can be formulated as:

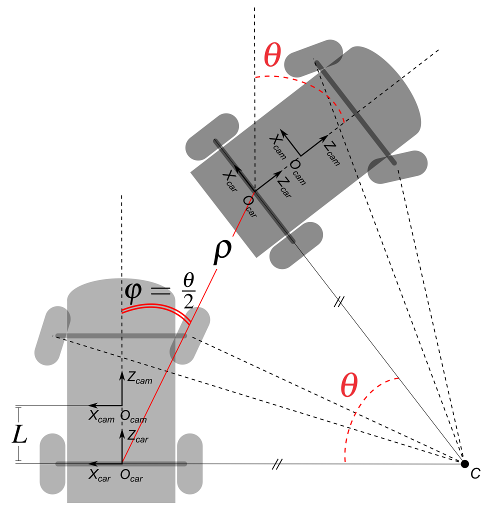


Fig. 2: Ackermann steering geometry: assuming locally circular and planar motion for wheeled vehicles, the vehicle motion can be recovered up to a scale factor by estimating a single angle.

$$\bar{\mathtt{R}}_1 = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \text{ and } \bar{\mathbf{t}}_1 = \rho \begin{bmatrix} \sin\varphi \\ 0 \\ \cos\varphi \end{bmatrix}, \quad (14)$$

where matrix $\bar{\mathtt{R}}_1$ contains the rotation increment, vector $\bar{\mathbf{t}}_1$ represents the translation, and ρ is the translation length.

Let $\mathscr{P}_1 \subset \mathscr{P}$ contain the points belonging to static parts of the scene. For every point $P_h \in \mathscr{P}_1$, the angular increment between two consecutive frames g and g' can be estimated by imposing the epipolar constraint [14]. So, for every 2D point correspondence h, we can write an equation as:

$$(\mathbf{w}_{g'h} \otimes \mathbf{w}_{gh})\, [\bar{\mathtt{T}}_1]_x \, \bar{\mathtt{R}}_1 = 0, \quad (15)$$

where matrix $[\bar{\mathtt{T}}_1]_x$ is the skew symmetric form of translation vector $\bar{\mathbf{t}}_1$, and operator $\otimes$ is the Kronecker product. For the case of planar and circular motion we have $\varphi = \frac{\theta}{2}$ in Eq. (14), and the motion can be recovered up to a scale factor ρ only by estimating the increment angle θ_h for point P_h, such that:

$$\theta_h = -2 \arctan \frac{v_{g'h}\, u_{gh} - u_{g'h}\, v_{gh}}{\lambda_{g'h}\, v_{gh} + v_{g'h}\, \lambda_{gh}}, \quad (16)$$

where u, v and λ are the components of vector $\mathbf{w}_{gh}$ as in Eq. (1).

The estimated angular increment for every point $P_i \in \mathscr{P}$ can be a hypothesis for the camera motion, but if we assume that the camera motion is the dominant perceived motion—which is the assumption of all visual navigation algorithms—then the hypothesis that is supported by more points is the camera ego-motion. In order to compensate the image measurement errors, a safety margin is considered for the estimated camera motion. That means, the angular motion increment for the camera is $\theta_{cam} = \theta \pm t_a$, where t_a is a threshold value.

Note that the motivation for estimating the camera motion is to segment the point correspondences that belong to static parts of the scene. Therefore, estimating the translation scale ρ is ignored and only the angle θ_{cam} is used to identify the inliers for estimating the camera motion. Later, the motion-segmentation hypotheses are evaluated and then 3D structures and the corresponding motions for the best hypothesis are estimated in Alg. (3).

B. Modeling Eoru-motion

The main moving objects that can be seen by a car while driving are cars, buses, bikes and pedestrians. In this work, we assume that all moving objects are rigid.

Planar motion of rigid objects is a more general case of circular and planar motion which is discussed in Section III-A. In case of planar motion, Eq. (14) holds but the constraint of circular motion is relaxed, so $\varphi \neq \frac{\theta}{2}$. Therefore, the planar motion has two degrees of freedom and at least two point correspondences are required to estimate the motion [32]. Considering Eq. (14), by estimating the two angles φ and θ the motion can be recovered up to translation scale ρ. So, writing the epipolar constraint (Eq. (15)) for two point correspondences results in two equations that are sufficient to estimate the two unknowns θ, and φ in Eq. (14).

Using the planar motion model for moving parts of the scene and the circular-planar motion model for stationary parts of the scene, several hypotheses are generated (see Alg. (2)) which are very fast to evaluate, thanks to minimal motion models.

C. Generating Hypotheses for Motion Segments

To segment p point trajectories into n motions, several hypotheses for such motion segmentation are generated and then each hypothesis is evaluated to find the best segmentation. The motion-segmentation hypotheses are evaluated and refined in an iterative process and then 3D structures and motions are estimated for the best hypothesis. Basically, a motion-segment hypothesis represents a possible partition of all point trajectories. A hypothesis is generated by first, selecting entries of trajectory matrix $\mathtt{W}$, which comply with the camera ego-motion model and removing them from the trajectory matrix. The remaining part of trajectory matrix $\mathtt{W}$, containing the entries that do not belong to the camera motion, is called $\overline{\mathtt{W}}$. Then, the planar motion model is used to segment matrix $\overline{\mathtt{W}}$ into $n-1$ other segments representing the eoru-motion. To find the remaining segments for every hypothesis—considering that matrix $\overline{\mathtt{W}}$ contains only planar motions—a sample pairs of columns from matrix $\overline{\mathtt{W}}$, which have moved similarly among the f frames, are selected. Then, by estimating motions for each sample, the remaining entries in matrix $\overline{\mathtt{W}}$ are evaluated to find the points that move the same way as each sample. This process is repeated with the reminders of matrix $\overline{\mathtt{W}}$ in a multi-RANSAC like scheme, as presented in [34]. In this way, a set of hypotheses for motion segments is generated from trajectory matrix $\mathtt{W}$, see Alg. (2).

In more detail, for each hypothesis, a pair of points from the set $\overline{\mathscr{P}} = \mathscr{P} - \mathscr{P}_1$ is selected, and using these points, a new trajectory matrix $\mathtt{W}_j^{(s)}$ is constructed from the entries of matrix $\overline{\mathtt{W}}$. Then, the rotation and translation for motion j are recovered using the 2-point algorithm [32].

Algorithm 2 Generating hypotheses for motion segmentation

Input: *2D image correspondences* ($\mathtt{W}$)
Output: *Several motion segmentation hypotheses* ($\mathtt{W}^c$)

for $c = 1$ to ψ **do**
 ▷ *% generate ψ hypotheses for motion segments%*
  $\mathtt{W}^c = \mathtt{W}_1$ ▷ *% initialize the segmented trajectory matrix %*
  $j = 2$ ▷ *% j represents the motion index%*
  while $\overline{\mathscr{P}} \neq \emptyset$ **do**
   while (reprojection error $> \varepsilon$) **do**
    ▷ *% reject invalid hypotheses%*
    Sample $k = 2$ points from set $\overline{\mathscr{P}}$ and form $\mathtt{W}_j^{(s)}$
    Estimate $\mathtt{M}_j$ and $\mathtt{S}_j^{(s)}$
    ▷ *% using epipolar geometry%*
    Calculate the reprojection error
   end while
   Estimate structure for $\overline{\mathtt{W}}$ with respect to $\mathtt{M}_j$
   Remove points from $\overline{\mathscr{P}}$ that comply with $\mathtt{M}_j$
   $\mathtt{W}^c \leftarrow [\mathtt{W}^c \mid \mathtt{W}_j]$
   ▷ *% add points from $\overline{\mathscr{P}}$ that comply with $\mathtt{M}_j$ to $\mathtt{W}^c$ %*
   $j = j + 1$
  end while
end for

The point correspondences in trajectory matrix $\mathtt{W}_j^{(s)}$ agree on a unique motion if reprojection error of the estimated structure is less than a threshold ε, such that:

$$\|\mathtt{W}_j^{(s)} - (\mathtt{M}_j \ \mathtt{S}_j^{(s)})\| < \varepsilon, \tag{17}$$

where matrix $\mathtt{S}_j^{(s)}$ is the estimated structure for the pair of points in $\mathtt{W}_j^{(s)}$. If Eq. (17) does not hold, sampling points from matrix $\overline{\mathtt{W}}$ continues until a pair of points that have a similar motion is identified.

Once a motion is identified, other points in set $\overline{\mathscr{P}}$ will be verified to check whether they comply with the identified motion using:

$$\mathtt{S}_{\bar{j}} = \mathtt{M}_j^{\top} \ \mathtt{W}_{\bar{j}}, \tag{18}$$

where matrices $\mathtt{S}_{\bar{j}}$ and $\mathtt{W}_{\bar{j}}$ represent 3D and 2D coordinates, respectively, of the points that are in set $\overline{\mathscr{P}} - \mathscr{P}_j$ and $\bar{j}$ is the index of points in set $\overline{\mathscr{P}} - \mathscr{P}_j$.

To generate a motion segmentation hypothesis, this process will be repeated until all points in $\overline{\mathscr{P}}$ (or the columns of trajectory matrix $\overline{\mathtt{W}}$) are associated to a motion-segment. Alg. (2) shows the process of generating ψ motion-segmentation hypotheses.

D. Evaluating Motion Segments' Hypotheses

From every hypothesis, an initial estimate of motion segments is generated. Given an initial motion segmentation for each hypothesis, it is possible to estimate the 3D structure and motion for each motion segment independently. The depth scales λ_{gi} are initialized recursively, as in Eq. (6). Such factorization can be done either on each motion segment

Algorithm 3 Evaluate hypotheses

Input: *Motion segmentation hypothesis*
Output: *Structures and motions*

for all motion hypotheses **do**
  for all motion segments **do**
    Estimate structure and motion
    Calculate reprojection error for each point
  end for
  repeat
    for all points with reprojection error $> \sigma$ **do**
      Add to another motion segment
      Triangulate new points in motion segments
      Calculate reprojection error
    end for
  until Convergence
end for
return Structures and Motions of the best hypothesis
$\triangleright$ *%with the smallest reprojection error%*

individually (via Eq. (7)) or on all motions and structures at the same time, using Eq. (11) as in [24] (see Appendix A). Then, the reprojection error for each point (e.g. point i under motion j) is calculated as below:

$$e_{ij} = \sum_{g=1}^{f} \left\| \mathbf{w}_{gi} - \tilde{\mathtt{M}}_{gj}\, \tilde{\mathbf{s}}_i \right\|^2 , \tag{19}$$

where vector $\tilde{\mathbf{s}}_i$ is the estimated position of point i in 3D space and matrix $\tilde{\mathtt{M}}_{gj}$ represents the estimated motion matrix that projects the points belonging to motion segment j on image frame g. Thus, those points that have the reprojection error larger than a certain threshold σ will be considered as outliers for that segment and called *segment-outliers*. In the next step, the 3D coordinates of segment-outliers are estimated under other motion segments and the corresponding reprojection error is calculated as in Eq. (19). This step continues until all the segment-outliers are assigned to another segment or rejected as global outliers. Finally, the estimated 3D structures and motions for the motion segmentation hypothesis, that results in the smallest reprojection error, are reported as the best solution. The process of evaluating motion segmentation hypotheses, and estimating motions and 3D structures for the winning hypothesis is outlined in Alg. (3).

IV. Experiments

To evaluate the performance of our method, a popular street-level dataset—*KITTI dataset*[1]—is used for the experiments. This dataset was originally created to benchmark VO algorithms [25]. It consists of several sequences collected by a perspective car-mounted camera driving in urban areas. Although the *KITTI dataset* provides stereo images, for our experiments the sequence from the left camera is used.

Since a 2-degree of freedom motion model is used for the on-road objects, only those that have planar rigid motion fit in this model. The case of pedestrians is not studied, because pedestrians motion cannot be considered as a rigid motion. Furthermore, in most cases, there are not sufficient and stable features on pedestrians to be considered as individual bodies. Pedestrian detection and tracking is out of the scope of this paper and there is a large literature on this topic. In this regard, a comprehensive study for monocular cameras is presented in [35] and also an effective method for driving-assistance systems is introduced by Enzweiler and Gavrila in [36].

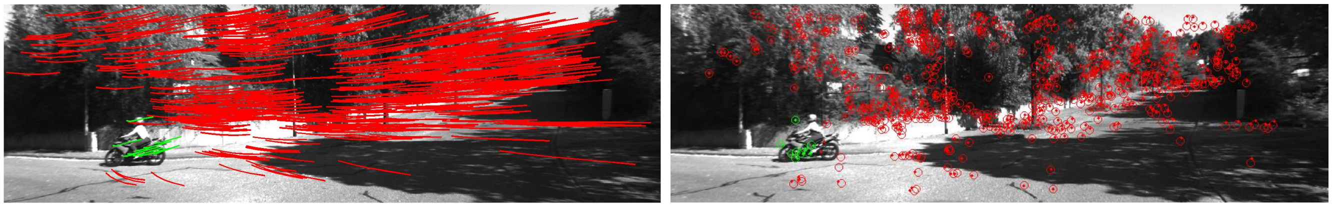
The input to our pipeline is the sequence of images. First, feature points are extracted from the images and matched between consecutive frames. In our experiments, SIFT features [37] are used and two-way matching is applied to each image pair. The feature matches are fed to the algorithm, which automatically rejects the outliers during the hypotheses' generation stage. The outputs of the algorithm are the estimated structures and motions.

Fig. 3 to Fig. 8 show four sample scenarios as well as the results obtained by our algorithm. As the sequences are from a car-mounted camera, in these figures the camera is moving forward and, consequently, the static parts of the scene are identified as an individual motion (in red). In these figures, the estimated motion is shown on the left and the reprojection error on the right. The detected feature points are shown as dots and the circles denote the projection of estimated 3D points on the images. Different colors—in both left and right subfigures—represent the estimated segmentation for point trajectories. In Fig. 3 the camera-equipped car is turning left while a motorcycle is moving almost perpendicularly to the camera motion. So, in addition to the ego-motion, the relative motion of the motorcycle is identified as eoru-motion. Another scenario is presented in Fig. 4, which shows a vehicle coming from the opposite direction while turning left. In this experiment, although there are some false negatives on the observed vehicle, most of the points are segmented correctly. Fig. 5 shows the case where a car is coming towards the camera, which generates a motion parallel to the camera motion and in the opposite direction. In Fig. 6 the car-mounted camera is moving forward and passing another car. The selected sequences contain scenarios with different relative types of ego-motion and eoru-motion, which are extracted by visually inspecting the whole dataset. The reprojection error for each point on every frame is calculated as $\|\mathbf{w}_{gi} - \tilde{\mathbf{w}}_{gi}\|$, and the segmentation error is defined as:

$$\text{Segmentation Error} = 100 \times \frac{\text{No. of misclassified points}}{\text{Total No. of points}}.$$

The motions and structures are estimated over a small window of image frames and used to initialize further frame windows over the sequence. In the experiments, the window size is considered to be $\mathscr{W} = 5$. Small number of frames allow the algorithm to have enough stable 2D correspondences among the frames, but the frame window should also be large enough to convey meaningful information about the motion. Fig. 3 to Fig. 6 show the motion trajectories over a frame window overlaid on the last frame of that particular frame window. The illustrated reprojection errors in these figures

[1] http://www.cvlibs.net/datasets/kitti/



(a) Estimated motions (b) Reprojection error

Fig. 3: Forward-Perpendicular(sequence *car_02_04*): The car-mounted camera is moving forward and a motorbike is driving perpendicularly to the car's motion.

(a) Estimated motions (b) Reprojection error

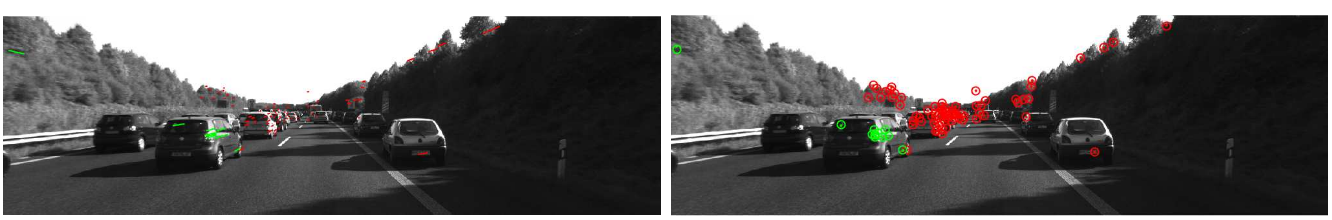
Fig. 4: Forward-Backward Curve (sequence *car_10_01*): The car-mounted camera is moving forward and another car is coming backward from the opposite direction and turning left.

(a) Estimated motions (b) Reprojection error

Fig. 5: Forward-Backward (sequence *car_11_01*): The car-mounted camera is moving forward while the other car is moving toward the camera.

(a) Estimated motions (b) Reprojection error

Fig. 6: Takeover (sequence *car_01_02*): The car-mounted camera is moving forward and passing another car which moves in the same direction.



(a) Estimated motions (b) Reprojection error

Fig. 7: Traffic jam sequence (sequence *car_20_01*): due to traffic jam, all cars but one are almost still and get classified as static part of the scene (red).

(a) Estimated motions (b) Reprojection error

Fig. 8: Traffic jam sequence (sequence *car_20_02*): due to traffic jam, all cars but one are almost still and get classified as static part of the scene (red).

correspond to the last frame in the window. Table I shows the mean reprojection errors among all frames for several sequences of the *KITTI dataset*, including those in Fig. 3 to Fig. 6. The segmentation error is defined as the percentage of points that are misclassified and, since the segmentation of the points is done over a frame window, this metric is measured for every frame window.

Fig. 7 and 8 depict a traffic jam, where most of the cars have very small (or in most cases do not have) relative apparent motion. For this reason, the features of all cars but one are classified as static scene (marked in red). We have chosen two temporally overlapping frames windows from *Sequence 20* of the *KITTI dataset* to investigate how the choice of frame windows affects the estimation of relative motion. In this experiment, frames 2 to 14 (Fig. 7) and 9 to 18 (Fig. 8) from this video sequence are considered as two individual frame windows and have 6 frames in common. As shown in Fig. 7 and Fig. 8, the feature points are segmented differently. This shows that motion segmentation is highly affected by the apparent relative motion in a frame window. Moreover, the resulted segmentation and reprojection errors—presented in Table I—also differ. This reflects the fact that the algorithm chooses the best segmentation given a frame window, which may not be the best segmentation for the next frame window.

We compare our proposed framework with state-of-the-art algorithms, such as [6], [38]–[44], on *car* sequences from the benchmark dataset *Hopkins 155*[2] [26]. This dataset was recorded with a hand-held camera. Therefore, minimal motion models could not be used; instead, a general motion model (as in [24]) is employed in our framework. Fig. 9 and Fig. 10 show two sample cases from the *car* sequence of *Hopkins 155* dataset. In these figures, the segmented motions are depicted on the left-hand side and the reprojection error of the estimated structures on the right-hand side. The estimated ego-motions are marked in red, while the eoru-motion in other colors. The average reprojection error computed with our method for this sequence is 0.091 pixels (compared with other methods in Table III). The reprojection error is shown in Fig. 9b and Fig. 10b for two sample sequences as the difference between dots and centroids of circles. Table II presents the reprojection and segmentation errors for individual samples of the *car* sequences from *Hopkins dataset*. Table III shows the motion-segmentation error obtained with our approach against other state-of-the-art methods [6], [38]–[44]. An overview of these methods is reported in [45]. Note that, the reprojection error for other methods is not available in Table III, because these methods are only used for motion segmentation and not for 3D reconstruction.

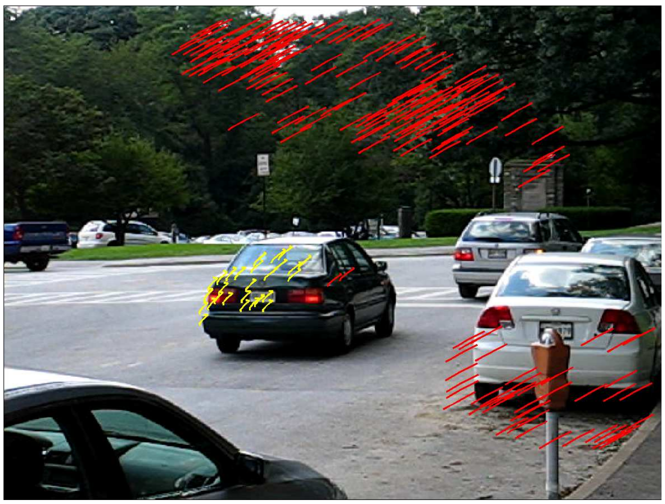
[2]http://www.vision.jhu.edu/data/hopkins155/

The proposed method has a few parameters to tune. The whole pipeline illustrated in Alg. (1)–(3) contains only three threshold values: the angular motion threshold t_a for estimating the camera ego-motion and two other thresholds on reprojection error, i.e. ε and σ, used for motion-segmentation hypothesis generation and evaluation, respectively. The choice of threshold t_a is discussed in [33]. In practice, we observed that $t_a = 2.5^\circ$ gives high-quality inliers for ego-motion and, as suggested in [33], the points are validated by checking the reprojection error being $\leq \sigma$. As we experienced, the choice of reprojection error threshold ε is not independent from the quality of feature-point detection and matching. The better the correspondences, the smaller the value that can be used for ε. Using smaller values for ε allows the method to fit more accurate models on the scene and estimate the motions with smaller reprojection error. Consequently, the more accurate the models, the lower the threshold σ to evaluate other points with respect to the model. In general, a smaller threshold (i.e. ε) is required to generate the model from the sampled point set compared to the one used for evaluating the points with respect to the model (i.e. σ). Thus, we have $\varepsilon < \sigma$. Comparing the reprojection error obtained for both *KITTI* and *Hopkins 155* datasets (illustrated in Table I and Table II), one can see that the reprojection error for the *KITTI* dataset is larger than the *Hopkins 155* dataset. This is because the *Hopkins 155* dataset comes with 2D point correspondences across the frames, and since the camera is hand-held, the ego-motion is relatively smaller in this dataset and more features remain visible and stable across the frames. Differently, for the *KITTI* dataset, feature points are extracted and matched without any post-processing to evaluate matches. Moreover, in the *KITTI* dataset, the camera is mounted on a car which is moving relatively fast and it results in losing the tracks of some features across the frames.

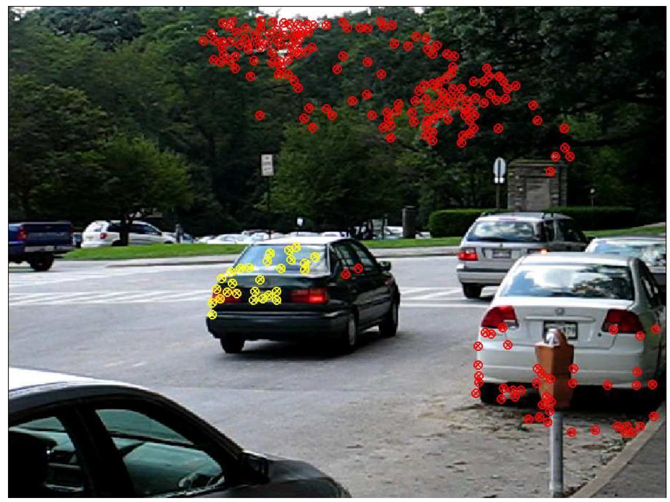
Finally, to show the influence of the type of motion models, our method is compared with our previous work [24], which uses general 6-DOF motions. Fig. 11 shows the number of required iterations per number of observable motions in the scene using either the general 6-DOF motion model or the locally planar and circular motion. As shown in Fig. 11 the number of iterations increases as more motions are observed

TABLE I: Reprojection and segmentation errors for sequences from *KITTI dataset*.

	Reprojection Error (pixels)	Segmentation Error (%)	Number of Motions	Number of Frames
car_10_01	0.217	0	2	8
car_08_08	0.086	0	2	8
car_08_04	0.171	0	2	5
car_11_01	1.29	0	2	13
car_02_04	0.609	0	2	9
car_01_02	0.423	0	2	20
car_20_01	0.113	1.81	2	10
car_20_02	0.069	0.65	2	13



(a) Estimated motions: ego-motion in *red* and eoru-motion in *yellow*.



(b) Reprojection error: 2D measurements are appeared as dots and back-projection of estimated 3D structure as circles.

Fig. 9: *car2* sequence from *Hopkins 155*: Motion trajectories and reprojection error for the last image in frame-window.

(a) Estimated motions: ego-motion in *red* and eoru-motion in *yellow* and *blue*.

(b) Reprojection error: 2D measurements are appeared as dots and back-projection of estimated 3D structure as circles.

Fig. 10: *car9* sequence from *Hopkins 155*: Motion trajectories and reprojection error for the last image over frame-windows.

TABLE II: Reprojection and segmentation errors for sequences from Hopkins 155 dataset.

	Reprojection Error (pixels)	Segmentation Error (%)	Number of Motions	Number of Frames
car1	0.177	0	2	20
car2	0.070	0	2	30
car4	0.043	0	2	50
car7	0.040	0	2	25
car8	0.063	0	2	22
car9	0.156	0	3	20
truck2	0.1347	0	2	22

TABLE III: Our proposed framework with generic 6-DOF motion model in comparison with the state-of-the-art methods on *car* sequence from *Hopkins 155*. The segmentation errors for sequences with two and three motions are shown separately.

	Reprojection Error (pixels)	Mean Segmentation Error (%) for two motions	Median Segmentation Error (%) for two motions	Mean Segmentation Error (%) for three motions	Median Segmentation Error (%) for three motions
Our Method	0.091	0	0	0.11	0.24
SSC [39]	-	1.20	0.32	0.52	0.28
GPCA [38]	-	1.41	0.00	19.83	19.55
LSA [6]	-	5.43	1.48	25.07	23.79
LLMC [40]	-	2.13	0.00	5.62	0.00
MSL [41]	-	2.23	0.00	1.80	0.00
ALC [42]	-	2.83	0.30	4.01	1.35
SLBF [43]	-	0.20	0.00	0.38	0.00
RANSAC [44]	-	2.55	0.21	12.83	11.45

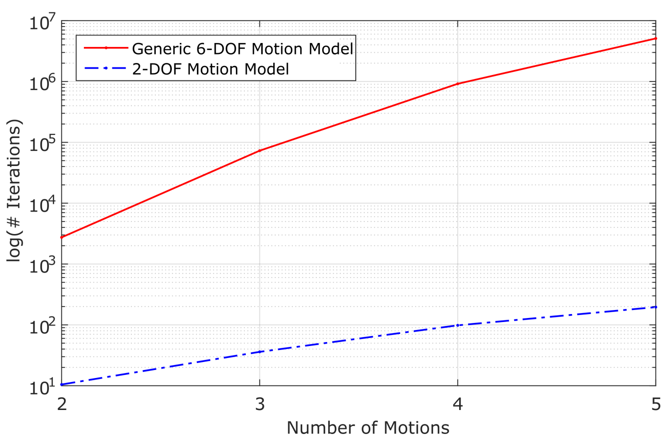


Fig. 11: The required number of iterations with respect to the numbers of eoru-motions in the scene.

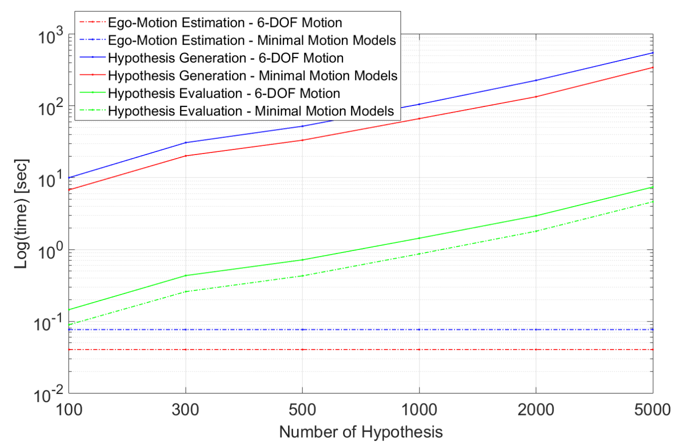


Fig. 12: The algorithm runtime against the number of generated hypothesis for both generic and minimal motion models.

by the camera. This fact is described by [34] with

$$K = \frac{\log(1-p)}{\log(1-\omega^n)},$$

where K is the number of required iterations for RANSAC model fitting process, p is the desired percentage of inliers in the selected set of points (which is $p = 0.99$), ω is the probability of inliers in the whole set of points and n is the number of points required to model the motion.

Comparing the use of unconstrained and constrained motion models (as shown in Fig. 11), the number of required iterations substantially decreases by enforcing the motion constraints. As extensively discussed in [33] and [14], using the constrained motion model instead of the unconstrained one may result in a drop in accuracy. In general, this drop in accuracy depends on how the ego-motion and eoru-motion diverge from the assumption of locally planar and circular motion. The validity of these assumptions is subject to the terrain, the camera frequency, and the speed of moving objects.

The proposed pipeline is composed of three major parts; ego-motion estimation, hypothesis generation for segmenting eoru-motions, and evaluation of eoru-motion segmentation hypotheses. Several runs of the algorithm on different sequences show that hypothesis generation is the most time-consuming part of the pipeline and takes almost 99% of the runtime, regardless of the motion model. Fig. 12 shows the average run-times of a Matlab implementation of the three major parts of the pipeline, with both generic (6-DOF) and minimal motion models. The presented runtimes in Fig. 12 are obtained from the Matlab implementation of the algorithm on a consumer laptop [3]. Thus, the computational performance can be improved by more efficient implementations, i.e. using C++ and parallel programming. Indeed, as the number of generated hypotheses increases, the runtime also increases. In most sequences of both *KITTI* and *Hopkins 155* datasets, a good motion segmentation could be found by generating at most 300 hypotheses. In order to compute the runtime, the pipeline was run on all the sequences with a fixed window size of $\mathscr{W} = 5$ and the reported runtime is the average of runtimes of all the sequences, given a specific number of hypotheses for eoru-motion segmentation.

Although the algorithm proposed in this paper does not have any theoretical limitation on the maximum number of motions that it can handle, Fig. 11 reveals that the maximum number of motions that the system can manage is limited by the available resources (i.e. memory and processing power) of the machine. It also shows that, since exploiting the motion models decreases the demand for resources, the ability of the system to manage more moving objects increases.

Comparing results of the proposed method on both *KITTI* and *Hopkins 155* datasets, the obtained reprojection error for the *KITTI* dataset is larger than the other one. This is because the *Hopkins 155* provides the point correspondences and the wrong matches are removed a priori. Differently, for the *KITTI* dataset, we provide the point correspondences and the wrong ones are not removed. Since the wrong matches do not comply with any motion in the scene, they will be ignored during different stages of our algorithm. However, such wrong and inaccurate correspondences cause imperfect motion segmentation and consequently less accurate estimated motions and structures.

The robustness of the algorithm is characterized by the segmentation error, which is the percentage of wrong point association to each motion segment. Such erroneous associations can be considered as outliers for the motion segmentation algorithm, which also increase the mean reprojection error. Note that the outliers in frame-to-frame point correspondences do not need to be detected or removed a priori. Using a conventional outlier removal method (i.e. RANSAC) is not an option, because it will remove the wrong correspondences together with points that correspond to other motions rather than the dominant motion. Such kind of outliers will be removed as the algorithm tries to fit multiple motion models to the set of point trajectories. However, outliers in feature correspondence are one of the major issues in all feature-based computer vision algorithms. In the context of localization and mapping, direct methods are being introduced to avoid relying on sparse features. Both methods have their own pros and cons, and the choice of using either of these methods strictly depends on the application.

The assumption that dominant observable motion corresponds to the camera ego-motion is valid only if the vehicle-mounted camera is not stationary. However, the absence of motion can be trivially detected by an IMU, wheel encoders, or by checking that features did not move from one frame to the other.

[3]Intel i7 - 2.6 GHz Processor, 16 GB RAM

V. CONCLUSION

This paper proposed a theoretical framework to simultaneously segment and estimate motions of multiple objects (called eoru-motion) and ego-motion of a camera. The framework was first derived for general, unconstrained motion and then adapted to vehicle-mounted cameras. The kinematics of the vehicle and of the other on-road moving objects are taken into account and used to speed up the process. The performance of our method was evaluated on a set of street-level sequences from a benchmark dataset. The results showed that our approach with minimal motion models can effectively perform in urban environments. Furthermore, comparing against the state-of-the-art motion segmentation methods on another benchmark dataset showed that our approach with general motion model performs successfully for this problem. Such improvement in the runtime of the algorithm—in comparison with our previous work [24]—supports the motivation for using motion constraints in this problem.

Possible extensions of this work are the inclusion of non-rigid motions, as well as the handling of occlusions and missing trajectories. In case of articulated motion, if enough points on each part of the moving object are tracked across the frames, the motions are segmented as independent motions and consequently each part is considered individually in the MBSfM pipeline.

APPENDIX A
SOLVING FOR MULTIPLE STRUCTURES AND MOTIONS

Once the matrices $\mathtt{W}$, $\mathtt{M}$ and $\mathtt{S}$ in Eq. (11) are formed, the estimations of structures and motion-segments are refined iteratively. This can be achieved by alternatively estimating the structures matrix $\tilde{\mathtt{S}}$ while fixing motions and estimating the motions matrix $\tilde{\mathtt{M}}$ while fixing structures, where matrices $\tilde{\mathtt{M}}$ and $\tilde{\mathtt{S}}$ are defined in Eq. (7).

Considering Eqs (7) and (11), given multiple motions matrix $\tilde{\mathtt{M}}$, estimation of multiple structures matrix $\tilde{\mathtt{S}}$ can be formalized as an optimization problem that solves a linear system of equations. In more detail, Eq. (7) can be rewritten in form of $\mathtt{A}\mathbf{x} = \mathbf{b}$, such as:

$$\widehat{\mathtt{M}}\ \overrightarrow{\mathtt{S}} = vec(\tilde{\mathtt{W}}), \tag{20}$$

where matrix $\widehat{\mathtt{M}} \in \mathbb{R}^{3fp\times 4np}$ contains $4np_j$ columns for every motion in a block-diagonal way, and is defined as:

$$\widehat{\mathtt{M}} = \begin{bmatrix} \widehat{\mathtt{M}}_1 & & & \\ & \widehat{\mathtt{M}}_2 & & \\ & & \ddots & \\ & & & \widehat{\mathtt{M}}_n \end{bmatrix}_{3fp\times 4np},$$

$$\widehat{\mathtt{M}}_j = \begin{bmatrix} \check{\mathtt{M}}_1^{(j)} & \check{\mathtt{M}}_2^{(j)} & \dots & \check{\mathtt{M}}_f^{(j)} \end{bmatrix}^\top_{3fp_j\times 4np_j}, \qquad j = 1 \dots n, \tag{21}$$

$$\check{\mathtt{M}}_g^{(j)} = \begin{bmatrix} \tilde{\mathtt{M}}_g & & \\ & \ddots & \\ & & \tilde{\mathtt{M}}_g \end{bmatrix}_{3p_j\times 4np_j}, \qquad g = 1 \dots f,$$

$$\tilde{\mathtt{M}}_g = \begin{bmatrix} \tilde{\mathtt{M}}_{g1} \mid \tilde{\mathtt{M}}_{g2} \mid \dots \mid \tilde{\mathtt{M}}_{gn} \end{bmatrix}, \qquad \tilde{\mathtt{M}}_g \in \mathbb{R}_{3\times 4n},$$

and $\overrightarrow{\mathtt{S}}$ is a column-wise vectorization of matrix $\tilde{\mathtt{S}}$, such that:

$$\overrightarrow{\mathtt{S}}_{4np\times 1} = \begin{bmatrix} \mathbf{s}_1 \; \mathbf{0}_a \; \mathbf{s}_2 \dots \mathbf{0}_a \; \mathbf{s}_{p_1} \; \mathbf{0}_{a'} \dots \mathbf{0}_a \; \mathbf{s}_{p_j} \; \mathbf{0}_{a'} \dots \mathbf{0}_a \; \mathbf{s}_{p_n} \end{bmatrix}^\top, \tag{22}$$

where $\mathbf{0}_a$ and $\mathbf{0}_{a'}$ are vectors of a and a' zeros, $a = 4(n-1)$ and $a' = 4n$. Finally, $vec(\tilde{\mathtt{W}})$ is the column-wise vectorization of $\tilde{\mathtt{W}}$. Structure of these matrices is shown in Fig. 13.

Now, we can solve Eq. (20) to estimate the structures. The equations belonging to non-zero values of $\overrightarrow{\mathtt{S}}$ can be used to create systems of equations to estimate structures in a *least-squares* sense. To that end, every non-zero block of $\overrightarrow{\mathtt{S}}$—representing a moving structure—forms an independent linear system of equations which can be solved individually. Note that, it is also possible to exploit the sparsity of vector $\overrightarrow{\mathtt{S}}$ as an additional constraint in the optimization process (as in [4]) and solve Eq. (20) for all the structures and motions simultaneously.

Once we have an estimate for the structure matrix $\mathtt{S}$, estimating the motions is possible by rewriting the Eq. (7) as:

$$(\tilde{\mathtt{S}}^\top \otimes \mathtt{I}_{3f}) \; vec(\tilde{\mathtt{M}}) = vec(\tilde{\mathtt{W}}), \tag{23}$$

where $\mathtt{I}_{3f}$ is a $3f \times 3f$ identity matrix and $vec(\tilde{\mathtt{M}})$ is column-wise vectorization of $\tilde{\mathtt{M}}$.

Using Eq. (20) and Eq. (23), we alternate between estimating multiple structures and multiple motions until they converge.

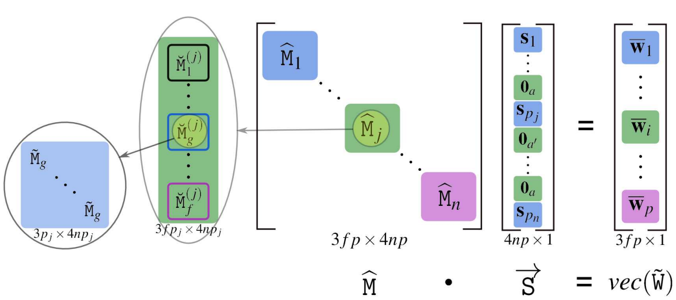


Fig. 13: Structure of matrices in Eq. (20) for p points having n motions in f frames.

References

[1] D. Scaramuzza and F. Fraundorfer, "Visual odometry: Part I: The first 30 years and fundamentals," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[2] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, "Collaborative monocular slam with multiple micro aerial vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'13)*. IEEE, 2013, pp. 3962–3970.

[3] Y. Murakami, T. Endo, Y. Ito, and N. Babaguchi, "Depth-estimation-free condition for projective factorization and its application to 3d reconstruction," in *Asian Conference on Computer Vision (ACCV'13)*. Springer, 2013, pp. 150–162.

[4] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *International Journal of Computer Vision (IJCV'98)*, vol. 29, no. 3, pp. 159–179, 1998.

[5] L. Zappella, A. Del Bue, X. Lladó, and J. Salvi, "Joint estimation of segmentation and structure from motion," *Computer Vision and Image Understanding (CVIU'13)*, vol. 117, no. 2, pp. 113–129, 2013.

[6] J. Yan and M. Pollefeys, "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate," in *European Conference on Computer Vision (ECCV'06)*. Springer, 2006, pp. 94–106.

[7] R. Vidal, Y. Ma, S. Soatto, and S. Sastry, "Two-view multibody structure from motion," *International Journal of Computer Vision (IJCV'06)*, vol. 68, no. 1, pp. 7–25, 2006.

[8] R. Vidal and R. Hartley, "Three-view multibody structure from motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI'08)*, vol. 30, no. 2, pp. 214–227, 2008.

[9] P. Ji, H. Li, M. Salzmann, and Y. Dai, "Robust motion segmentation with unknown correspondences," in *European Conference on Computer Vision (ECCV'14)*. Springer, 2014, pp. 204–219.

[10] K. Schindler, U. James, and H. Wang, "Perspective n-view multibody structure-and-motion through model selection," in *European Conference on Computer Vision (ECCV'06)*. Springer, 2006, pp. 606–619.

[11] T. Li, V. Kallem, D. Singaraju, and R. Vidal, "Projective factorization of multiple rigid-body motions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*. IEEE, 2007, pp. 1–6.

[12] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part II: Matching, robustness, optimization, and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

[13] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, "Real-time monocular visual odometry for on-road vehicles with 1-point ransac," in *IEEE International Conference on Robotics and Automation (ICRA'09)*. IEEE, 2009, pp. 4293–4299.

[14] D. Scaramuzza, "1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints," *International Journal of Computer Vision (IJCV'11)*, vol. 95, no. 1, pp. 74–85, 2011.

[15] P. Simionescu and D. Beale, "Optimum synthesis of the four-bar function generator in its symmetric embodiment: the ackermann steering linkage," *Mechanism and Machine Theory*, vol. 37, no. 12, pp. 1487–1504, 2002.

[16] D. Scaramuzza, A. Censi, and K. Daniilidis, "Exploiting motion priors in visual odometry for vehicle-mounted cameras with non-holonomic constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'11)*. IEEE, 2011, pp. 4469–4476.

[17] Y. Jiang, H. Chen, G. Xiong, and D. Scaramuzza, "Icp stereo visual odometry for wheeled vehicles based on a 1dof motion prior," in *IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, 2014*, 2014.

[18] R. Vidal, "Multi-subspace methods for motion segmentation from affine, perspective and central panoramic cameras," in *IEEE International Conference on Robotics and Automation (ICRA'05)*. IEEE, 2005, pp. 1216–1221.

[19] A. Kundu, K. M. Krishna, and C. Jawahar, "Realtime multibody visual slam with a smoothly moving monocular camera," in *IEEE International Conference on Computer Vision (ICCV'11)*. IEEE, 2011, pp. 2080–2087.

[20] C. Vogel, S. Roth, and K. Schindler, "View-consistent 3d scene flow estimation over multiple frames," in *European Conference on Computer Vision (ECCV'14)*. Springer, 2014, pp. 263–278.

[21] C. Rabe, U. Franke, and R. Koch, "Dense 3d motion field estimation from a moving observer in real time," in *5th Biennial Workshop on DSP for In-Vehicle Systems*, 2011, pp. 1–8.

[22] D. Pfeiffer and U. Franke, "Modeling dynamic 3d environments by means of the stixel world," *Intelligent Transportation Systems Magazine (ITS'11)*, vol. 3, no. 3, pp. 24–36, 2011.

[23] H. Badino and T. Kanade, "A head-wearable short-baseline stereo system for the simultaneous estimation of structure and motion," in *12th IAPR Conference on Machine Vision Applications*, 2011, pp. 185–189.

[24] R. Sabzevari and D. Scaramuzza, "Monocular simultaneous multi-body motion segmentation and reconstruction from perspective views," in *IEEE International Conference on Robotics and Automation (ICRA'14)*. IEEE, 2014, pp. 23–30.

[25] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*. IEEE, 2012, pp. 3354–3361.

[26] R. Tron and R. Vidal, "A benchmark for the comparison of 3-d motion segmentation algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*. IEEE, 2007, pp. 1–8.

[27] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision (IJCV'92)*, vol. 9, no. 2, pp. 137–154, 1992.

[28] P. Sturm and B. Triggs, "A factorization based algorithm for multi-image projective structure and motion," in *European Conference on Computer Vision (ECCV'96)*. Springer, 1996, pp. 709–720.

[29] R. Truesdalc, K. T. M. Micropaloeonr, and J. Fenner, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, p. 133, 1981.

[30] J. Oliensis and R. Hartley, "Iterative extensions of the sturm/triggs algorithm: Convergence and nonconvergence," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI'07)*, vol. 29, no. 12, pp. 2217–2233, 2007.

[31] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robotos - Second Edition*. The MIT press, 2011.

[32] D. Ortin and J. Montiel, "Indoor robot motion based on monocular images," *Robotica*, vol. 19, no. 3, pp. 331–342, 2001.

[33] D. Scaramuzza, "Performance evaluation of 1-point-ransac visual odometry," *Journal of Field Robotics*, vol. 28, no. 5, pp. 792–811, 2011.

[34] M. Zuliani, C. S. Kenney, and B. Manjunath, "The multiransac algorithm and its application to detect planar homographies," in *IEEE International Conference on Image Processing (ICIP'05)*, vol. 3. IEEE, 2005, pp. 150–153.

[35] M. Enzweiler and D. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI'09)*, vol. 31, no. 12, pp. 2179–2195, 2009.

[36] M. Enzweiler and Gavrila, "A multilevel mixture-of-experts framework for pedestrian classification," *IEEE Transactions on Image Processing (TIP'11)*, vol. 20, no. 10, pp. 2967–2979, 2011.

[37] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV'04)*, vol. 60, no. 2, pp. 91–110, 2004.

[38] Y. Ma, A. Y. Yang, H. Derksen, and R. Fossum, "Estimation of subspace arrangements with applications in modeling and segmenting mixed data," *SIAM review*, vol. 50, no. 3, pp. 413–458, 2008.

[39] E. Elhamifar and R. Vidal, "Clustering disjoint subspaces via sparse representation," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP'10)*. IEEE, 2010, pp. 1926–1929.

[40] A. Goh and R. Vidal, "Segmenting motions of different types by unsupervised manifold clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*. IEEE, 2007, pp. 1–6.

[41] Y. Sugaya and K. Kanatani, "Geometric structure of degeneracy for multi-body motion segmentation," in *Statistical Methods in Video Processing*. Springer, 2004, pp. 13–25.

[42] Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of multivariate mixed data via lossy data coding and compression," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI'07)*, vol. 29, no. 9, pp. 1546–1562, 2007.

[43] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, "Hybrid linear modeling via local best-fit flats," *International Journal of Computer Vision (IJCV'12)*, vol. 100, no. 3, pp. 217–240, 2012.

[44] A. Y. Yang, S. R. Rao, and Y. Ma, "Robust statistical estimation and segmentation of multiple subspaces," in *Computer Vision and Pattern Recognition Workshop (CVPRW'06)*. IEEE, 2006, pp. 99–99.

[45] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.

Reza Sabzevari is a postdoctoral researcher at the Robotics and Perception Group (RPG), University of Zurich, Switzerland. He received his PhD in Computer Vision from the Italian Institute of Technology (IIT) and the University of Genoa, Italy, in 2013. His research interests include multiple-view geometry, photometric stereo and active perception. He conducted his research through prestigious research grants from Hasler Foundation and UZH Forschungskredit.

Davide Scaramuzza (1980, Italian) is Professor of Robotics at the University of Zurich, where he does research at the intersection of robotics, computer vision, and neuroscience. He did his PhD in robotics and computer Vision at ETH Zurich and a postdoc at the University of Pennsylvania. From 2009 to 2012, he led the European project sFly, which introduced the world's first autonomous navigation of micro drones in GPS-denied environments using vision as the main sensor modality. For his research contributions, he was awarded an SNSF-ERC Starting Grant, the IEEE Robotics and Automation Early Career Award, and a Google Research Award. He coauthored the book Introduction to Autonomous Mobile Robots (published by MIT Press).